THE AGGREGATORS

Vijeta, Prasanna, Aditi, Behrooz, Zachary

03 March, 2021

Introduction- Road accident has become very common these days. In USA more than 38,000 people die in road crash each year and on an average 4.4 million people are injured seriously enough to require medical attention. Source- <https://www.asirt.org/safe-travel/road-safety-facts/> Looking at the severity of the accidents, for this project we have decided to use the US Accident- A Countrywide Traffic Accident Dataset (2016-2000) to find something helpful.

Dataset Information-

The dataset is available at the Kaggle Portal(<https://www.kaggle.com/sobhanmoosavi/us-accidents>).This is a country wide traffic accident data-set which covers 49 states of the US. There are about 4.2 million accident records in this data-set.It contains all sort of information related to each accident like the weather condition during the accident, which side the accident occurred, address, time of accident etc. There are a total of 49 observations.

Below mentioned are the description of Variable

ID: A unique identifier of accident record Source: Types of API who reported Accidents TMC: Traffic message channel code which defines the situation in detail. Severity: Indicates effect of accident on road traffic by number between 1 and 4. Where 1 indicates short delays and 4 indicates Long delays Start\_time: Shows start time of the accident in the local time zone. End\_Time: Shows end time of the accident in the local time zone.  
Start\_Lat: Shows latitude in GPS coordinate of the start point Start\_Lng: Shows longitude in GPS coordinate of the start point. End\_Lat: Shows latitude in GPS coordinate of the end point. End\_Lng: Shows longitude in GPS coordinate of the end point. Distance(mi): The distance on the road affected by the accident. Description: Accident’s senario is explained in words. Number: Specifies the street number in the address variable. Street: Indicates the street name in the address column. Side: Shows the relative side of the street (Right/Left) in address field City: Indicates the city in the address field. County: Depicts the county in the address field. State: Shows the state in the address column. Zipcode: Shows the zipcode in the address field. Country: Shows the country in the address field. Timezone: Indicates time zone depending on the accidents location Airport\_Code: Denotes an airport-based weather station which is the closest accident’s location.. Weather\_Timestamp: Specify the time-stamp of the weather observation record (in local time). Temperature(F): Shows the temperature (in Fahrenheit). Wind\_Chill(F): Shows the wind chill (in Fahrenheit). Humidity(%): Shows the humidity (in percentage). Pressure(in):Shows the air pressure (in inches). Visibility(mi): Shows visibility (in miles). Wind\_Direction: Shows wind direction. Wind\_Speed(mph): Shows wind speed (in miles per hour). Precipitation(in): Indicates precipitation amount in inches, if there is any Weather\_Condition: Specifies the type of Weather (rain, snow, thunderstorm, fog, etc. Amenity: A POI annotation which indicates existence of amenity in a surrounding location. Bump: A POI annotation which specifies presence of speed bump or hump in surrounding accident location. Crossing:Indicates presence of crossing in a nearby location. Give\_Way: Specifies presence of give\_way in a nearby location. Junction: Denotes presence of junction in a at location of accidents No\_Exit: Indicates existence of no\_exit at location of accident. Railway: Implies existence of railway track or crossing  
Roundabout: existence of roundabout at location of accident Station: Shows the presence of a station in a surrounding incident location. Stop: presence of stop in a nearby location. Traffic\_Calming: A POI annotation which indicates presence of traffic\_calming in a nearby location. Traffic\_Signal: A POI annotation which indicates presence of traffic\_signal within a few miles. Turning\_Loop: Specifies the presence of turning\_loop in a nearby location. Sunrise\_Sunset: Indicates the time of day (i.e. day or night) based on sunrise/sunset. Civil\_Twilight: Shows the period of day (i.e. day or night) based on civil twilight. Nautical\_Twilight: Shows the period of day (i.e. day or night) based on nautical twilight Astronomical\_Twilight Shows the period of day (i.e. day or night) based on astronomical twilight.

Research Questions We will try to find the trend of accident and the factors effecting accidents. We will also analyze the details of US Accidents in different states to be able to check what can be done to reduce accidents and also to avoid its effect on road traffic.

#run library  
  
library(tidyverse)

Import dataset

US\_Accidents <- read\_csv("US\_Accidents.csv")

List column names of data set

colnames(US\_Accidents)

## [1] "ID" "Source" "TMC"   
## [4] "Severity" "Start\_Time" "End\_Time"   
## [7] "Start\_Lat" "Start\_Lng" "End\_Lat"   
## [10] "End\_Lng" "Distance(mi)" "Description"   
## [13] "Number" "Street" "Side"   
## [16] "City" "County" "State"   
## [19] "Zipcode" "Country" "Timezone"   
## [22] "Airport\_Code" "Weather\_Timestamp" "Temperature(F)"   
## [25] "Wind\_Chill(F)" "Humidity(%)" "Pressure(in)"   
## [28] "Visibility(mi)" "Wind\_Direction" "Wind\_Speed(mph)"   
## [31] "Precipitation(in)" "Weather\_Condition" "Amenity"   
## [34] "Bump" "Crossing" "Give\_Way"   
## [37] "Junction" "No\_Exit" "Railway"   
## [40] "Roundabout" "Station" "Stop"   
## [43] "Traffic\_Calming" "Traffic\_Signal" "Turning\_Loop"   
## [46] "Sunrise\_Sunset" "Civil\_Twilight" "Nautical\_Twilight"   
## [49] "Astronomical\_Twilight"

Number of rows in data-set

nrow(US\_Accidents)

## [1] 4232541

Sampling US\_Accidents data-set. The new data-set will contain 1 million rows

US\_Accidents\_Sample <- US\_Accidents[sample(nrow(US\_Accidents), 1000000, replace = FALSE, prob = NULL),]

Checking type of data frame to confirm it is a tibble

str(US\_Accidents\_Sample)

## tibble [1,000,000 x 49] (S3: tbl\_df/tbl/data.frame)  
## $ ID : chr [1:1000000] "A-1777987" "A-249758" "A-3806977" "A-2844486" ...  
## $ Source : chr [1:1000000] "MapQuest" "MapQuest" "Bing" "Bing" ...  
## $ TMC : num [1:1000000] 201 201 NA NA 201 201 NA NA 241 201 ...  
## $ Severity : num [1:1000000] 2 2 2 2 2 3 2 2 3 2 ...  
## $ Start\_Time : POSIXct[1:1000000], format: "2018-11-22 16:58:53" "2016-10-12 19:52:23" ...  
## $ End\_Time : POSIXct[1:1000000], format: "2018-11-22 17:28:27" "2016-10-12 20:21:50" ...  
## $ Start\_Lat : num [1:1000000] 35.3 39 33.9 37.5 29.8 ...  
## $ Start\_Lng : num [1:1000000] -80.8 -76.9 -118.1 -122.1 -95.4 ...  
## $ End\_Lat : logi [1:1000000] NA NA NA NA NA NA ...  
## $ End\_Lng : logi [1:1000000] NA NA NA NA NA NA ...  
## $ Distance(mi) : num [1:1000000] 0 0.01 0 0.071 0 ...  
## $ Description : chr [1:1000000] "Accident on Cinderella Rd at Kentbrook Dr." "Left lane closed and left hand shoulder closed due to accident on MD-295 Baltimore Washington Pkwy Northbound a"| \_\_truncated\_\_ "At CA-42/Firestone Blvd - Accident." "At CA-109/University Ave - Accident." ...  
## $ Number : num [1:1000000] 699 NA NA NA 1696 ...  
## $ Street : chr [1:1000000] "Kentbrook Dr" "Baltimore Washington Pkwy N" "I-605 N" "Bayfront Expy W" ...  
## $ Side : chr [1:1000000] "L" "R" "R" "R" ...  
## $ City : chr [1:1000000] "Charlotte" "Lanham" "Norwalk" "Menlo Park" ...  
## $ County : chr [1:1000000] "Mecklenburg" "Prince George's" "Los Angeles" "San Mateo" ...  
## $ State : chr [1:1000000] "NC" "MD" "CA" "CA" ...  
## $ Zipcode : chr [1:1000000] "28213-6715" "20706" "90650" "94025" ...  
## $ Country : chr [1:1000000] "US" "US" "US" "US" ...  
## $ Timezone : chr [1:1000000] "US/Eastern" "US/Eastern" "US/Pacific" "US/Pacific" ...  
## $ Airport\_Code : chr [1:1000000] "KJQF" "KCGS" "KFUL" "KPAO" ...  
## $ Weather\_Timestamp : POSIXct[1:1000000], format: "2018-11-22 16:50:00" "2016-10-12 19:59:00" ...  
## $ Temperature(F) : num [1:1000000] 55.4 53.6 75 59 68 33.1 54 53 66 74 ...  
## $ Wind\_Chill(F) : num [1:1000000] NA NA 75 NA NA NA 54 53 NA 74 ...  
## $ Humidity(%) : num [1:1000000] 47 94 37 88 88 72 71 83 50 68 ...  
## $ Pressure(in) : num [1:1000000] 30.4 30.2 29.7 29.9 29.8 ...  
## $ Visibility(mi) : num [1:1000000] 10 10 10 10 8 10 7 5 10 10 ...  
## $ Wind\_Direction : chr [1:1000000] "Calm" "Calm" "W" "Calm" ...  
## $ Wind\_Speed(mph) : num [1:1000000] NA NA 5 NA 4.6 NA 7 13 6.9 8 ...  
## $ Precipitation(in) : num [1:1000000] NA NA 0 NA NA NA 0 0.11 NA 0 ...  
## $ Weather\_Condition : chr [1:1000000] "Clear" "Clear" "Fair" "Clear" ...  
## $ Amenity : logi [1:1000000] FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Bump : logi [1:1000000] FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Crossing : logi [1:1000000] FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Give\_Way : logi [1:1000000] FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Junction : logi [1:1000000] FALSE TRUE FALSE FALSE FALSE TRUE ...  
## $ No\_Exit : logi [1:1000000] FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Railway : logi [1:1000000] FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Roundabout : logi [1:1000000] FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Station : logi [1:1000000] FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Stop : logi [1:1000000] FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Traffic\_Calming : logi [1:1000000] FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Traffic\_Signal : logi [1:1000000] FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Turning\_Loop : logi [1:1000000] FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Sunrise\_Sunset : chr [1:1000000] "Day" "Night" "Night" "Day" ...  
## $ Civil\_Twilight : chr [1:1000000] "Day" "Night" "Night" "Day" ...  
## $ Nautical\_Twilight : chr [1:1000000] "Day" "Night" "Day" "Day" ...  
## $ Astronomical\_Twilight: chr [1:1000000] "Day" "Day" "Day" "Day" ...  
## - attr(\*, "problems")= tibble [3,032,128 x 5] (S3: tbl\_df/tbl/data.frame)  
## ..$ row : int [1:3032128] 2716478 2716478 2716479 2716479 2716480 2716480 2716481 2716481 2716482 2716482 ...  
## ..$ col : chr [1:3032128] "End\_Lat" "End\_Lng" "End\_Lat" "End\_Lng" ...  
## ..$ expected: chr [1:3032128] "1/0/T/F/TRUE/FALSE" "1/0/T/F/TRUE/FALSE" "1/0/T/F/TRUE/FALSE" "1/0/T/F/TRUE/FALSE" ...  
## ..$ actual : chr [1:3032128] "40.11206" "-83.03187" "39.86501" "-84.04873" ...  
## ..$ file : chr [1:3032128] "'US\_Accidents.csv'" "'US\_Accidents.csv'" "'US\_Accidents.csv'" "'US\_Accidents.csv'" ...

View sample dataset in new window and check column names and datatype

View(US\_Accidents\_Sample)# to view compelete dataset.  
  
glimpse(US\_Accidents\_Sample) # to check datatype of each column

## Rows: 1,000,000  
## Columns: 49  
## $ ID <chr> "A-1777987", "A-249758", "A-3806977", "A-2844...  
## $ Source <chr> "MapQuest", "MapQuest", "Bing", "Bing", "MapQ...  
## $ TMC <dbl> 201, 201, NA, NA, 201, 201, NA, NA, 241, 201,...  
## $ Severity <dbl> 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, 3, 2, 3, ...  
## $ Start\_Time <dttm> 2018-11-22 16:58:53, 2016-10-12 19:52:23, 20...  
## $ End\_Time <dttm> 2018-11-22 17:28:27, 2016-10-12 20:21:50, 20...  
## $ Start\_Lat <dbl> 35.26602, 38.99142, 33.92228, 37.48718, 29.77...  
## $ Start\_Lng <dbl> -80.78529, -76.88610, -118.10432, -122.14194,...  
## $ End\_Lat <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...  
## $ End\_Lng <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...  
## $ `Distance(mi)` <dbl> 0.000, 0.010, 0.000, 0.071, 0.000, 0.000, 1.1...  
## $ Description <chr> "Accident on Cinderella Rd at Kentbrook Dr.",...  
## $ Number <dbl> 699, NA, NA, NA, 1696, NA, NA, NA, NA, NA, NA...  
## $ Street <chr> "Kentbrook Dr", "Baltimore Washington Pkwy N"...  
## $ Side <chr> "L", "R", "R", "R", "R", "R", "R", "R", "R", ...  
## $ City <chr> "Charlotte", "Lanham", "Norwalk", "Menlo Park...  
## $ County <chr> "Mecklenburg", "Prince George's", "Los Angele...  
## $ State <chr> "NC", "MD", "CA", "CA", "TX", "CA", "MD", "CA...  
## $ Zipcode <chr> "28213-6715", "20706", "90650", "94025", "770...  
## $ Country <chr> "US", "US", "US", "US", "US", "US", "US", "US...  
## $ Timezone <chr> "US/Eastern", "US/Eastern", "US/Pacific", "US...  
## $ Airport\_Code <chr> "KJQF", "KCGS", "KFUL", "KPAO", "KMCJ", "KRDD...  
## $ Weather\_Timestamp <dttm> 2018-11-22 16:50:00, 2016-10-12 19:59:00, 20...  
## $ `Temperature(F)` <dbl> 55.4, 53.6, 75.0, 59.0, 68.0, 33.1, 54.0, 53....  
## $ `Wind\_Chill(F)` <dbl> NA, NA, 75, NA, NA, NA, 54, 53, NA, 74, 81, N...  
## $ `Humidity(%)` <dbl> 47, 94, 37, 88, 88, 72, 71, 83, 50, 68, 65, 2...  
## $ `Pressure(in)` <dbl> 30.35, 30.21, 29.69, 29.89, 29.78, 29.99, 30....  
## $ `Visibility(mi)` <dbl> 10, 10, 10, 10, 8, 10, 7, 5, 10, 10, 10, 10, ...  
## $ Wind\_Direction <chr> "Calm", "Calm", "W", "Calm", "Variable", "Cal...  
## $ `Wind\_Speed(mph)` <dbl> NA, NA, 5.0, NA, 4.6, NA, 7.0, 13.0, 6.9, 8.0...  
## $ `Precipitation(in)` <dbl> NA, NA, 0.00, NA, NA, NA, 0.00, 0.11, NA, 0.0...  
## $ Weather\_Condition <chr> "Clear", "Clear", "Fair", "Clear", "Clear", "...  
## $ Amenity <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...  
## $ Bump <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...  
## $ Crossing <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...  
## $ Give\_Way <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...  
## $ Junction <lgl> FALSE, TRUE, FALSE, FALSE, FALSE, TRUE, FALSE...  
## $ No\_Exit <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...  
## $ Railway <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...  
## $ Roundabout <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...  
## $ Station <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...  
## $ Stop <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...  
## $ Traffic\_Calming <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...  
## $ Traffic\_Signal <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...  
## $ Turning\_Loop <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...  
## $ Sunrise\_Sunset <chr> "Day", "Night", "Night", "Day", "Day", "Night...  
## $ Civil\_Twilight <chr> "Day", "Night", "Night", "Day", "Day", "Night...  
## $ Nautical\_Twilight <chr> "Day", "Night", "Day", "Day", "Day", "Night",...  
## $ Astronomical\_Twilight <chr> "Day", "Day", "Day", "Day", "Day", "Night", "...

colnames(US\_Accidents\_Sample) # to check column names

## [1] "ID" "Source" "TMC"   
## [4] "Severity" "Start\_Time" "End\_Time"   
## [7] "Start\_Lat" "Start\_Lng" "End\_Lat"   
## [10] "End\_Lng" "Distance(mi)" "Description"   
## [13] "Number" "Street" "Side"   
## [16] "City" "County" "State"   
## [19] "Zipcode" "Country" "Timezone"   
## [22] "Airport\_Code" "Weather\_Timestamp" "Temperature(F)"   
## [25] "Wind\_Chill(F)" "Humidity(%)" "Pressure(in)"   
## [28] "Visibility(mi)" "Wind\_Direction" "Wind\_Speed(mph)"   
## [31] "Precipitation(in)" "Weather\_Condition" "Amenity"   
## [34] "Bump" "Crossing" "Give\_Way"   
## [37] "Junction" "No\_Exit" "Railway"   
## [40] "Roundabout" "Station" "Stop"   
## [43] "Traffic\_Calming" "Traffic\_Signal" "Turning\_Loop"   
## [46] "Sunrise\_Sunset" "Civil\_Twilight" "Nautical\_Twilight"   
## [49] "Astronomical\_Twilight"

View Unique States in the sample data

unique(US\_Accidents\_Sample$State)

## [1] "NC" "MD" "CA" "TX" "NJ" "AZ" "PA" "WA" "NY" "IL" "FL" "CT" "OH" "OR" "RI"  
## [16] "VA" "SC" "AL" "TN" "MN" "MI" "GA" "OK" "CO" "IN" "UT" "LA" "WI" "MA" "DC"  
## [31] "MO" "MS" "NE" "IA" "NV" "DE" "KS" "ID" "NM" "KY" "AR" "WV" "NH" "ME" "MT"  
## [46] "WY" "ND" "VT" "SD"

Export sample data to share with Team members

write\_csv(US\_Accidents\_Sample,"US\_Accidents\_Sample.csv")

Tidy data-set- This dataset has 1 million rows and it has many anomalies like the date and time format, multiple values for zipcode in the same row, etc. To address all these tidying data is an important step to be followed. Following code tidy up our data set.

#Changing the Start\_Time date format and converting into default format  
  
US\_Accidents\_Sample$Start\_Time <- as.POSIXct(US\_Accidents\_Sample$Start\_Time , format = '%Y/%m/%d %H:%M:%S', tz = 'UTC')

#Adding weekday column to the dataset for analysis  
  
US\_Accidents\_Sample$weekday <- weekdays(US\_Accidents\_Sample$Start\_Time)

#Separating the time and date value for "Start\_time" and "End\_time" Column using separate function.  
  
US\_Accidents\_Sample <- US\_Accidents\_Sample %>%   
 separate(Start\_Time, into= c("Accident\_year","Start\_Accident\_month", "Start\_Accident\_date", "Start\_Accident\_Hour","Start\_Accident\_min","Start\_Accident\_sec")) %>% separate(End\_Time, into= c("End\_Accident\_year","End\_Accident\_month", "End\_Accident\_date", "End\_Accident\_Hour","End\_Accident\_min","End\_Accident\_sec"))

Removing columns which are not required for our analysis. To know the column names or number, we first check the column name and then with select function we remove the unwanted columns. Assigning the changes to the same dataset (US\_Accidents\_Sample) to avoid confusion.

colnames(US\_Accidents\_Sample)

## [1] "ID" "Source" "TMC"   
## [4] "Severity" "Accident\_year" "Start\_Accident\_month"   
## [7] "Start\_Accident\_date" "Start\_Accident\_Hour" "Start\_Accident\_min"   
## [10] "Start\_Accident\_sec" "End\_Accident\_year" "End\_Accident\_month"   
## [13] "End\_Accident\_date" "End\_Accident\_Hour" "End\_Accident\_min"   
## [16] "End\_Accident\_sec" "Start\_Lat" "Start\_Lng"   
## [19] "End\_Lat" "End\_Lng" "Distance(mi)"   
## [22] "Description" "Number" "Street"   
## [25] "Side" "City" "County"   
## [28] "State" "Zipcode" "Country"   
## [31] "Timezone" "Airport\_Code" "Weather\_Timestamp"   
## [34] "Temperature(F)" "Wind\_Chill(F)" "Humidity(%)"   
## [37] "Pressure(in)" "Visibility(mi)" "Wind\_Direction"   
## [40] "Wind\_Speed(mph)" "Precipitation(in)" "Weather\_Condition"   
## [43] "Amenity" "Bump" "Crossing"   
## [46] "Give\_Way" "Junction" "No\_Exit"   
## [49] "Railway" "Roundabout" "Station"   
## [52] "Stop" "Traffic\_Calming" "Traffic\_Signal"   
## [55] "Turning\_Loop" "Sunrise\_Sunset" "Civil\_Twilight"   
## [58] "Nautical\_Twilight" "Astronomical\_Twilight" "weekday"

US\_Accidents\_Sample <- US\_Accidents\_Sample %>%   
select(-c(2,3,19,20,29,30,32,33,55,57,58,59))

Changing few column names to make them more readable.

colnames(US\_Accidents\_Sample) = c("ID","Severity", "Accident\_year", "Start\_Accident\_month","Start\_Accident\_date","Start\_Accident\_hour","Start\_Accident\_min", "Start\_Accident\_sec","End\_Accident\_year","End\_Accident\_month","End\_Accident\_date","End\_Accident\_Hour","End\_Accident\_min","End\_Accident\_sec","Start\_Lat","Start\_Lng","Distance","Description","Number","Street","Side","City","County","State","Timezone","Temperature","Wind\_Chill","Humidity","Pressure","Visibility","Wind\_Direction","Wind\_Speed","Precipitation","Weather\_Condition","Amenity","Bump","Crossing","Give\_Way","Junction","No\_Exit","Railway","Roundabout","Station","Stop","Traffic\_Calming","Traffic\_Signal","Sunrise\_Sunset", "Weekday")

Converting few columns to correct datatypes.

US\_Accidents\_Sample <- US\_Accidents\_Sample %>%   
 mutate\_at(c(3,4,5,6,7,8,9,10,11,12,13,14), as.integer) %>%   
 mutate\_at(c(2), as.factor)

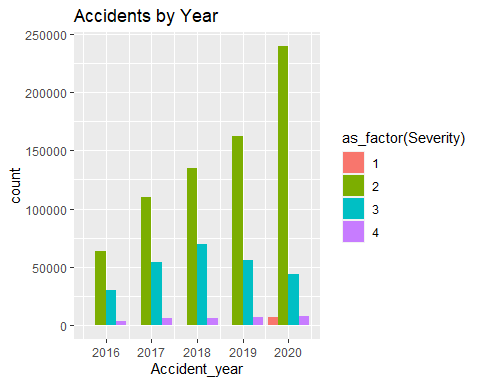
After tidying up, adding new columns, changing datatype and changing column names of the dataset, the new US\_Accidents\_Sample has the following column names.

view(US\_Accidents\_Sample)  
colnames(US\_Accidents\_Sample)

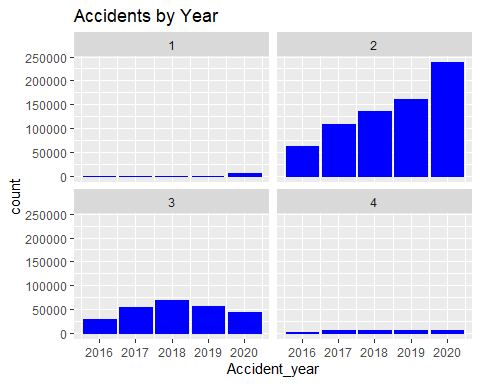
## [1] "ID" "Severity" "Accident\_year"   
## [4] "Start\_Accident\_month" "Start\_Accident\_date" "Start\_Accident\_hour"   
## [7] "Start\_Accident\_min" "Start\_Accident\_sec" "End\_Accident\_year"   
## [10] "End\_Accident\_month" "End\_Accident\_date" "End\_Accident\_Hour"   
## [13] "End\_Accident\_min" "End\_Accident\_sec" "Start\_Lat"   
## [16] "Start\_Lng" "Distance" "Description"   
## [19] "Number" "Street" "Side"   
## [22] "City" "County" "State"   
## [25] "Timezone" "Temperature" "Wind\_Chill"   
## [28] "Humidity" "Pressure" "Visibility"   
## [31] "Wind\_Direction" "Wind\_Speed" "Precipitation"   
## [34] "Weather\_Condition" "Amenity" "Bump"   
## [37] "Crossing" "Give\_Way" "Junction"   
## [40] "No\_Exit" "Railway" "Roundabout"   
## [43] "Station" "Stop" "Traffic\_Calming"   
## [46] "Traffic\_Signal" "Sunrise\_Sunset" "Weekday"

Accident Per Year- The following query calculates and visualizes the number of accidents per year.

by\_year<- US\_Accidents\_Sample %>%   
 group\_by(Accident\_year) %>%   
 summarise(total\_accident= n())  
  
#To group the number of accidents based on year we used group\_by function for Accident\_year, and then used summarise function to calculate total number of accidents.  
  
ggplot(data = US\_Accidents\_Sample)+  
 geom\_bar(mapping = aes(x= Accident\_year, fill= as\_factor(Severity)), position = "dodge")+  
 ggtitle("Accidents by Year")+  
 theme\_grey()



#using ggplot we created a bar graph , adding a layer to ggplot using geom\_bar and in ascetics we have taken accident\_year on X-Axis and fill for severity, followed by position function with dodge to align them side by side without any overlapping of bars, then titled the graph as Accidents by Year using ggtitle function, then gave the theme in light color for background using theme\_light  
  
#Since in the above graph severity 1 and 4 accidents were not clearly visible, we plotted separate graph for each severity type.  
  
ggplot(data = US\_Accidents\_Sample)+  
 geom\_bar(mapping = aes(x= Accident\_year,colour = "Severity"), colour = "blue", fill = "blue")+  
 ggtitle("Accidents by Year")+  
 theme\_grey()+  
 facet\_wrap(~Severity)

 Observations- From the above bar graph we can see that the total number of accidents are gradually increasing with every passing year. The bar graph shows that year 2020 has most number of accidents, although the number of severity 3 accidents has declined after 2018. It also shows that majority of the accidents belong to severity 2. Severity 1 and 4 accidents are rare and have been consistent in number over years.

Accident by State, arranged in descending order and visualizing top 5 states by number of accidents.

by\_state<- US\_Accidents\_Sample %>%   
 group\_by(State) %>%   
 summarise(No.of\_accident = n()) %>%   
 arrange(desc(No.of\_accident))  
  
# Here to get the data by state, we have grouped data by state then summarized it to get the count of accidents by state and arranged it in descending order.  
   
#Summary of accidents by state   
summary(by\_state)

## State No.of\_accident   
## Length:49 Min. : 54   
## Class :character 1st Qu.: 1966   
## Mode :character Median : 10362   
## Mean : 20408   
## 3rd Qu.: 24936   
## Max. :229893

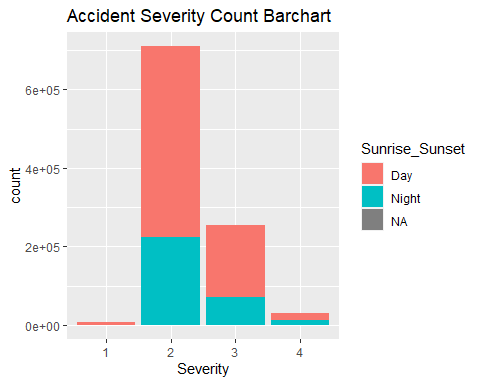
#graph showing top 5 states with highest number of accidents.  
  
top\_states<- top\_n(by\_state, 5)  
  
# Here we have used gglot to plot the top 5 states with highest number of accidents.  
  
  
ggplot(data= top\_states, aes(x=State, y=No.of\_accident)) +  
 geom\_histogram(stat="identity", fill="steelblue")+  
 ggtitle("Top 5 Accident States")+  
 geom\_text(aes(label=percentage\_accident), vjust=1.5, color="white", size=3.5)+  
 theme\_classic()

## Error in FUN(X[[i]], ...): object 'percentage\_accident' not found

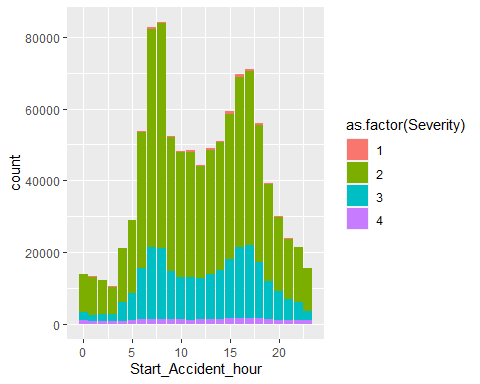
Observations- From the graph we can conclude that California has highest number of accidents with almost 23% of all accidents in USA.The next state after California is Texas with approx 9% of the total number of accidents. It would be interesting to know what factors contribute sharp decline in number of accidents in Texas considering that TX is twice the size of CA.

Analyzing the distribution of severity of the accidents and checking at which time majority of these accidents occur.

ggplot(data= US\_Accidents\_Sample, mapping = aes(x= Severity))+  
 geom\_bar(mapping = aes(fill = Sunrise\_Sunset))+  
 ggtitle("Accident Severity Count Barchart")



#The function ggplot() creates a coordinate system in which you can add layers to it and geom\_bar(data=top\_states) adds a blank graph, The mapping argument is always paired with aes(), and the x and y arguments of aes() specify which variables to map to the x and y axis and geom\_bar() will create a bar chart with x axis = Severity  
  
#As now we know that day has majority of accidents, now we are trying to find which time of the day has most accidents.  
  
by\_timeofDay<-  
 US\_Accidents\_Sample %>%   
 group\_by(Start\_Accident\_hour) %>%   
 summarise(accident\_count= n())  
  
#Here we have used group\_by() to group Start\_Accident\_hour ,summarise() will Collapse many values down to a single summary into variable named count.The Pipe %>% operator is used to update a value by first piping it into one or more expressions, and then assigning the result.  
  
ggplot(data = US\_Accidents\_Sample)+  
 geom\_bar(mapping = aes(x=Start\_Accident\_hour, fill = as.factor(Severity)))

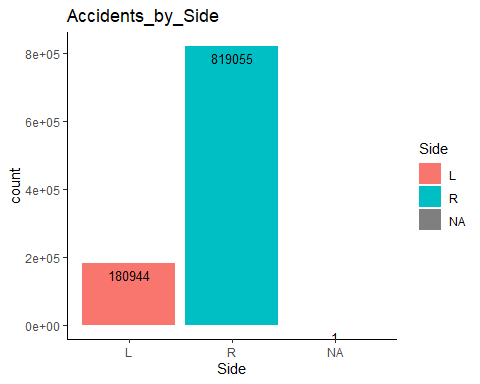


#The function ggplot() creates a coordinate system that you can add layers to and ggplot(data=US\_Accident\_hour) adds a blank graph. The next layer is geom which determines the shape of the chart, here it is bar chart with x= Start\_Accident\_hour and as Severity has character datatype to change it in categorical data so we used as.factor

Observation- From the bar graph we can see that most of the accidents occurs in day time. One possible reason that we could think of is due to higher commute during day compared to night. From the second graph, we can see that most of the accidents occurs at 7am, 8am, 4pm and 5pm. One of the reasons could be these are office starting hours and most of the people leave for work and come back from work at these timings.

Analyze which side of the road has more accidents.

by\_side <- US\_Accidents\_Sample %>%   
 group\_by(Side) %>%   
 summarise(count = n())  
  
#Here we have used group\_by() to group Side variable,summarise() will Collapse many values down to a single summary into variable named count.The Pipe %>% operator is used to update a value by first piping it into one or more expressions, and then assigning the result.  
  
  
 ggplot(data = by\_side, mapping = aes (x= Side,y= count, fill = Side))+  
 geom\_bar(stat="identity")+  
 ggtitle("Accidents\_by\_Side")+  
 geom\_text(aes(label= count), vjust=1.5, color="black", size=3.5)+  
 theme\_classic()



#The function ggplot() creates a coordinate system that you can add layers to and ggplot(data=by\_side) adds a blank graph. The next layer is geom which determines the shape of the chart, here it is bar chart. ggtitle is used to give over all title to chart and geom\_text is used to modify text like label is used to display count on bars.

Observations- The graph shows that most of the accidents occurs at the right side of the road whereas left side are very less number of accidents. It is quite surprising because the left most lanes is the fastest one. One reason for this could be high number of lane merging that vehicles do while entering or exiting the express.

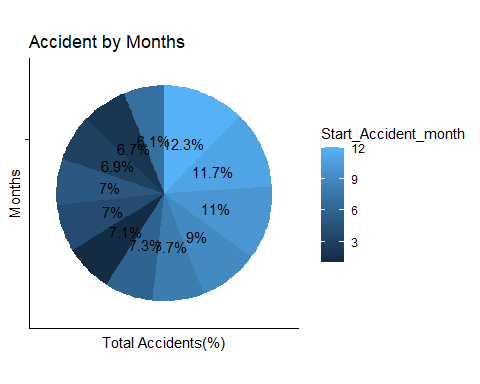
Analyze number of accidents in different Weather conditions along with to 5 weather conditions that contributes the most.

by\_weather\_condition<- US\_Accidents\_Sample %>%   
 group\_by(Weather\_Condition) %>%   
 summarise(No\_of\_Accident=n()) %>%   
 arrange(desc(No\_of\_Accident))  
  
#Top 10 weather conditions for most of the accidents  
  
top5\_WC<- top\_n(by\_weather\_condition, 5)

Observation- Counter intuitively, most of the accidents appear to occur in fair and clear weather conditions.

Analyze which month has the highest number of accidents.

by\_month<- US\_Accidents\_Sample %>%  
group\_by(Start\_Accident\_month) %>%  
summarise(No.of\_accident = n()) %>%  
arrange(desc(No.of\_accident)) %>%   
mutate(percentage\_accident = round(No.of\_accident/sum(No.of\_accident)\*100,2))  
  
#Here we have used group by to group Start\_Accident\_month than assigned the total number of accidents in a variable. We also created a new column by using mutate(), in this we used round() which rounds the values in its first argument to the specified number of decimal places(default 0).sum() in percentage formula is used to get values in percentage.  
  
ggplot(data=by\_month, mapping=aes(x="",y=percentage\_accident,fill = Start\_Accident\_month)) +  
geom\_bar(stat="identity")+  
ggtitle("Accident by Months")+  
geom\_text(aes(label=paste0(round(percentage\_accident,1),"%")), position = position\_stack(vjust=0.5)) +  
coord\_polar("y",start=0)+  
theme\_classic()+  
theme(axis.text = element\_blank()) + xlab("Months") + ylab("Total Accidents(%)")



#The function ggplot() creates a coordinate system that you can add layers to and geom\_bar(data=by\_month) adds a blank graph. Geom is another layer which determines the shape of the chart.Pie Chart = stacked bar chart + polar coordinates. We have first created a bar chart using geom\_bar, stat = "identity" is to skip the aggregation and it will consider the y values and ggtitle is for adding title to overall plot title. geom\_text is for adding labels in which paste0 function in R simply concatenates the vector without any separator. A theme\_classic() shows x and y axis lines and no gridlines and the coord\_polar() is used to produce a pie chart.

Observation- This chart shows that July has the least percentage of accidents and December has the highest percentage of accidents which is the holiday season in the USA, it is quite surprising that holidays have an significant effect on accidents.

Preliminary Conclusion-

- Even though the number of low severity and very high severity accidents are almost constant over the years. The severity 2 accidents are sharply increasing resulting in higher total number of accidents.

- CA has the highest number of accidents as much as 2.5 times that of TX which is the second in the rank. It would be interesting to know what factors contribute sharp decline in number of accidents in Texas considering that TX is twice the size of CA.

- Most of the accidents happen during daytime possibly due to high traffic caused by people commuting to the office and back home.

- Even though left most lanes is the fastest one, the graph shows that right side of the lane has highest number of accidents. One reason for this could be high number of lane merging that vehicles do while entering or exiting the express. - Most of the accidents appear to occur in fair and clear weather conditions. It would be interesting to explore the severity of these accidents. - December being the holiday season has the highest number of accidents.